REMARKS/ARGUMENTS

Claims 1-28 are pending in the present application. Claims 3, 5, 10, 14, 15, 19, 21, 24 and 26 have been amended herewith. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 101

The Examiner rejected Claims 26-28 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

Applicants have amended Claim 26 in response thereto, and thus it is urged that the rejection of Claim 26 (and similarly for dependent Claims 27 and 28) under 35 U.S.C. § 101 has been overcome.

II. 35 U.S.C. § 102, Anticipation

The Examiner rejected Claims 1-6, 9, 11, 13, 14, 16-22, 24 and 26-28 under 35 U.S.C. § 102 as being anticipated by Maxwell, III et al. (U.S. Patent Number 6,973,417 hereafter referred to as Maxwell). This rejection is respectfully traversed.

Generally speaking, Claim 1 is directed to a methodology that provides access indicators for instructions and data that allows for examination of such indicators to determine code coverage with respect to such instructions and data by the processor without special instrumentation of code. The teachings of Maxwell explicitly require special code instrumentation, as will be now described in detail.

Specifically with respect to Claim 1, such claim recites a processor that performs two functions – it executes an instruction in the code, and it changes a state of an access indicator when the instruction is executed such that coverage data is generated during execution of the code by the processor. In contrast, the teachings of the Maxwell are fundamentally different, as Maxwell teaches that a target program is *simulated in a different data processing system* in order to allow for testing of the target program before the target hardware is in existence (col. 2, lines 6-8). In order to accomplish Maxwell's stated objective of estimated code execution time using this different data processing system, the source code, which has been instrumented, is actually compiled twice – once using a target compiler (Figure 1, element 20) to generate target assembler files, and again using a host compiler (Figure 1, element 38) to generate host executable code. Importantly, the target assembler files used by the assembler analyzer are not executed at all, but instead are merely analyzed and used to calculate an estimated execution time (col. 13, line 59 – col. 14, line 4). The code that is actually executed - the host system executable 40 - is executed by host system 30 (Figure 1, element 30; col. 14, lines 5-51) and this host system which executes the instructions does not update or change the state of the assembly analyzer 24 or function data table 26. Rather, this host system merely *accesses* information/data that is maintained in the function data table 26, such data having previously been generated/created by the assembly analyzer 24 (col. 5, lines 40-67). Importantly, this assembly analyzer does not execute target assembler files 22, but merely analyzes these files (col. 5, lines48-50; col. 13, lines 61-62). Thus, there is no teaching of *a processor that performs the two functions* expressly recited in Claim 1 – a processor that both (1) executes an instruction in the code, and (2) *changes a state of the access indicator* when the instruction is executed such that coverage data is generated during execution of the code by the processor. Rather, the cited reference teaches two distinct systems, where one processes target assembler files – but does not execute such files – in order to create data in a function data table by an assembly analyzer, and another system which actually executes the host executable on a host system but this host system that executes such code does not change a state of any type of access indicator. Accordingly, as every element of Claim 1 is not identically shown in a single reference, it is urged that Claim 1 is not anticipated by the cited reference[1].

In addition, because the Maxwell architecture is fundamentally different in that it simulates a target program on a different system than the target hardware, there would have been no motivation to modify Maxwell's teachings in accordance with the missing claimed features identified above, as such modification would essentially eviscerate the entire expressed purpose of being able to simulate target code on another system prior to the target hardware being available (Maxwell col. 2, lines 6-8). It is thus further urged that in addition to not being anticipated by Maxwell, Claim 1 is also not obvious in view of Maxwell[2].

Applicants initially traverse the rejection of Claims 2-6, 9 and 11 for reasons given above with respect to Claim 1 (of which Claims 2-6, 9 and 11 depend upon).

Further with respect to Claim 2 (and dependent Claims 3 and 5), such claim recites "receiving a signal at an instruction cache in the processor from a processor unit in the processor; and responsive to receiving the signal, changing the state of the access indicator by the instruction cache". As can be seen, Claim 2 is directed to particular features pertaining to an instruction cache. In particular, a signal is received at the instruction cache from a processor unit, and *the state of the access indicator is changed by this instruction cache that received the signal.* In rejecting these aspects of Claim 2, the Examiner cites Maxwell col. 4, lines 6-10 and Figure 1, element 24. Applicants urge that there is no type of operational co-action between the instruction cache described by Maxwell col. 4 and the assembler analyzer described by Maxwell Figure 1, element 24 (which is alleged to be equivalent to the claimed 'access

[1] For a prior art reference to anticipate in terms of 35 U.S.C. 102, every element of the claimed invention must be identically shown in a single reference. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

[2] The fact that a prior art device could be modified so as to produce the claimed device is not a basis for an obviousness rejection unless the prior art suggested the desirability of such a modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984). Although a device may be capable of being modified to run the way [the patent applicant's] apparatus is claimed, there must be a suggestion or motivation *in the reference* to do so. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990).

indicator'). Thus, the Maxwell instruction cache cannot update the assembler analyzer (which is alleged to be equivalent to the claimed 'access indicator'). Rather, the Maxwell instruction cache maintains an instruction cache image that keeps a record of the contents of an instruction cache *of the simulated hardware system*. As described above with respect to Claim 1, this simulated hardware system does not change or otherwise update the state of any type of access indicator, either by the instruction cache (as explicitly claimed in Claim 2), or otherwise. Quite simply, the Maxwell instruction cache does not perform any such change or update of the state of an access indicator. Rather, it merely maintains a record of the contents of the simulated hardware system's instruction cache (Maxwell col. 4, lines 5-10; col. 5, lines 16-19, col. 15, lines 25-30 and 50-60; and col. 16, lines 11-15 and 33-39). Therefore, as there are additional claimed features not identically shown in a single reference, it is further urged that Claim 2 has been erroneously rejected by the Examiner. Dependent Claims 3 and 5 are similarly allowable for these same further reasons given herewith with respect to Claim 2 (of which Claims 3 and 5 depend upon).

Further with respect to Claim 3, Applicants have amended such claim to recite additional features not taught by the cited reference. In particular, Claim 3 has been amended in accordance with the Specification description at page 25, lines 6-22 to recite that there are multiple execution units for which instructions are dispatched for execution. These claimed features advantageously allow for out-of-order and speculative execution of instructions (Specification page 26, lines 10-23). The Maxwell reference does not teach or otherwise contemplate such a system architecture or its resulting advantages, and therefore it is further urged that Claim 3 is not anticipated by the cited reference.

Further with respect to Claim 4, such claim recites "marking selected instructions in the code for generating the coverage data by associating access indicators with selected instructions in the code". As can be seen, Claim 4 is directed to a particular way for generating the coverage data, where the actual (selected) instructions in the code are marked. In rejecting Claim 4, the Examiner cites Maxwell's teaching at col. 5, lines 54-67 as teaching these claimed features of Claim 4. Applicants respectfully submit that this passage instead describes a function table and data contained therein. While this data may contain *statistics <u>about</u> code*, this function table and associated data does not include actual instructions in the code that are executed, and therefore this table does not describe any type of access indicators or *the marking of <u>actual</u> instructions of code that is <u>executed</u>*. Thus, it is further shown that Claim 4 is not anticipated by the cited reference.

Further with respect to Claim 5, such claim recites "wherein instructions in the instruction cache are located in different positions within the instruction cache and wherein the signal includes an identification of a position in the instruction cache for the instruction". As can be seen, this claim does not merely recite that the instructions in the cache are located in different positions, but goes further and recites that *the signal that is received at the instruction cache (per Claim 2) includes an identification of a position in the instruction cache for the instruction.* In rejecting Claim 5, the Examiner cites Maxwell's teaching at Figure 2B, element 50 as teaching such signal with such identifier. Applicants urge that, instead, this element 50 of Figure 2B is a processor block that provides hardware parameters 45 to connect to Host function 56 (Maxwell col. 14, lines 5-110). These hardware parameters include parameters such as CPU cycle time, cache size and memory access time (Maxwell col. 14, lines 13-19). They do not provide any type of position information in an instruction cache for an instruction. It is therefore further urged that Claim 5 has been erroneously rejected under 35 USC 102 as every element of the claimed invention is not identically shown in a single reference.

Further with respect to Claim 6, such claim recites "wherein the access indicator is located in a field in the instruction". This access indicator is defined, per independent Claim 1, to be an indicator associated with a given instruction whose state is changed when this given instruction is executed. Thus, per the features of Claim 6, when an instruction is executed, an access indicator associated with the instruction that is executed has its state changed, and this access indicator that has its state changed when the instruction is executed is itself located in a field of this same instruction that is executed to prompt the state change of the access indicator. *In effect, a field within the instruction itself is modified, when the instruction is executed, to change the state of an access indicator.* In rejecting Claim 6, the Examiner cites Maxwell col. 5, lines 54-67 as teaching this claimed feature. Applicants urge that while this data is associated with code segments, this data (1) is not changed when an instruction is executed, and (2) this data is not located in a field of the instruction that gets executed. Thus, for at least these two reasons alone, this cited passage does not teach the claimed feature of "wherein the access indicator is located in a field in the instruction". Accordingly, it is further urged that Claim 6 is not anticipated by the cited reference as every element of the claimed invention is not identically shown in a single reference.

With respect to Claim 13, such claim recites "an instruction cache, wherein the instruction cache receives instructions and marks an instruction as executed in response to detecting a signal indicating that the instruction has been executed; and a processor unit, wherein the processor unit generates the signal when the instruction has completed execution". As can be seen, the instruction cache performs two distinct operations – the instruction cache (1) receives instructions and (2) marks an instruction as executed in response to detecting a signal indicating that the instruction has been executed. In rejecting this aspect of Claim 13, the Examiner cites Maxwell's teaching of an instruction cache at col. 4, lines 6-10 and depicted in Figure 2B, element 58 as describing each of the instruction cache features recited in Claim 13. Applicants urge error in such assertion, as this cited passage merely describes an instruction cache image that keeps a record of the contents of an instruction cache of a simulated processor, where the system updates the instruction cache image after a program segment is executed. This operation is further described by Maxwell at col. 5, lines 16-19, col. 15, lines 25-30 and 50-60, and col. 16, lines 11-15 and 33-39. As can be seen by reviewing these additional passages in addition to the Examiner cited passage, the cache 'update' that is described by Maxwell is a loading of a new code segment such that the instruction cache object is able to calculate instruction access time portion of the total execution time calculation (col. 15, lines 24-30 and 50-60). This instruction cache *does not <u>mark an instruction</u> as executed in response to detecting a signal* indicating that the instruction has been executed. This instruction cache object 58 also (1) maintains an image of an instruction cache, as loaded into it by the system, and (2) calculates the instruction access time portion of the total execution time (col. 15, lines 24-30). Again, *this instruction cache object does not <u>mark an instruction as executed</u>*, as expressly recited in Claim 13. Thus, as every element recited in Claim 13 is not identically shown in a single reference, it is urged that Claim 13 is not anticipated by the cited reference.

Applicants traverse the rejection of Claims 14 and 16 for reasons given above with respect to Claim 13 (of which Claims 14 and 16 depend upon).

Applicants traverse the rejection of Claim 17 (and dependent Claims 18-22) for similar reasons to those given above with respect to Claim 1.

Applicants further traverse the rejection of Claims 18 (and dependent Claims 19 and 21) and 27 for similar reasons to the further reasons given above with respect to Claim 2.

Applicants further traverse the rejection of Claim 19 for similar reasons to the further reasons given above with respect to Claim 3.

Applicants further traverse the rejection of Claim 20 for similar reasons to the further reasons given above with respect to Claim 4.

Applicants further traverse the rejection of Claims 21 and 28 for similar reasons to the further reasons given above with respect to Claim 5.

Applicants further traverse the rejection of Claim 22 for similar reasons to the further reasons given above with respect to Claim 6.

With respect to Claim 24, such claim has been amended and it is urged that Claim 24 is allowable similar to Claims 7 and 8.

With respect to Claim 26 (and dependent Claims 27 and 28) such claim has been amended and it is urged that Claim 26 is allowable similar to Claim 10.

Therefore, the rejection of Claims 1-6, 9, 11, 13, 14, 16-22, 24 and 26-28 under 35 U.S.C. § 102 has been overcome.

III. Objected Claims

Claims 7, 8, 10, 12, 15, 23 and 25 were objected to as being dependent upon a rejected base claim, but were otherwise indicated as being allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claim. Claims 10 and 15 has been amended accordingly. It is urged that Claims 7, 8, 12, 23 and 25, in addition to being objected to but allowable if rewritten in independent form, are further allowable as being dependent upon an allowable independent claim, as described above with respect to independent Claims 1 and 17.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited reference and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: December 14, 2006

Respectfully submitted,

/Wayne P. Bailey/

Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants